Copyright

by

Thomas Edwin Cuthbertson

2015

The Report committee for Thomas Edwin Cuthbertson
certifies that this is the approved version of the following report:

Modeling climate variables using Bayesian finite mixture models

APPROVED BY

SUPERVISING COMMITTEE

Timothy H. Keitt, Supervisor

Peter Müller

Modeling climate variables using Bayesian finite mixture models

by

Thomas Edwin Cuthbertson, B.S.

Report

Presented to the Faculty of the Graduate School

of the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Statistics

The University of Texas at Austin

May 2015

Modeling climate variables using Bayesian finite mixture models

by

Thomas Edwin Cuthbertson, MSStat

The University of Texas at Austin, 2015

SUPERVISOR: Timothy H Keitt

This paper presents an alternative to point-based clustering models using a Bayesian finite mixture model. Using a simulation of soil moisture data in the Amazon region of South America, a Bayesian mixture of regressions is used to preserve periodic behavior within clusters. The mixture model provides a full probabilistic description of all uncertainties in the parameters that generated the data in addition to a clustering algorithm which better preserves the periodic nature of data at a particular pixel.

Contents

Chapter One: Introduction

When a set of data is believed to have been generated from different underlying processes, it can be thought of as a mixture of homogeneous subsets of the data. Controlling for the differences is essential to its modeling; if data is pooled, only the average overall behavior can be estimated. In many instances, there is no explicit indication of how the data should be grouped. Clustering is a method that allows for the estimation of the groupings of data with similar features. A good clustering algorithm is one that maximizes within-cluster similarity and minimizes between-cluster similarity. Similarity can be thought of as the inverse of distance (Rai, 2011, p. 2-3). In short, different clustering algorithms are defined by their measure of distance. Unfortunately, there is no one prevailing algorithm; the choice depends on the structure of the data.

Motivation

Phenology is the biological study of the timing of events. In the paper "Continental-scale patterns of *Cecropia* reproductive phenology: evidence from herbarium specimens" (Zaramea et al., 2011), the reproductive patterns of the Neotropical pioneer trees, *Cecropia*, are

analyzed by correlating the semi-annual blooming patterns with longitudinal precipitation and temperature data. In this paper, Fourier spectral analysis is used to identify the significant sinusoidal periodicity of these phenological events. Using principle component analysis (PCA) on the significant spectra and k-means classification, annual patterns are distinguished from sub-annual patterns. The boundaries of 9 climactic regions of Central America, the Caribbean, and South America are determined using PCA and 9-means clustering on average yearly temperature and precipitation values within 20km pixels. With Fourier co-spectral analysis, the covariation between the periodic variation of phenology and regional climate patterns is identified.

This study seeks to determine if clustering on longitudinal climactic patterns provides different groupings than clustering on overall averages. By clustering the average monthly values at a particular pixel, the periodic nature of bioclimactic observations may be better preserved when grouping pixels as compared to grouping by a single average value. With a common periodic nature and level in a given cluster, inference on the correlation to other events may be more accurate. As a preliminary benchmarking measure, Figure 1 shows the clustering of the soil moisture dataset from this analysis using the same methodology from the Zaramea study.

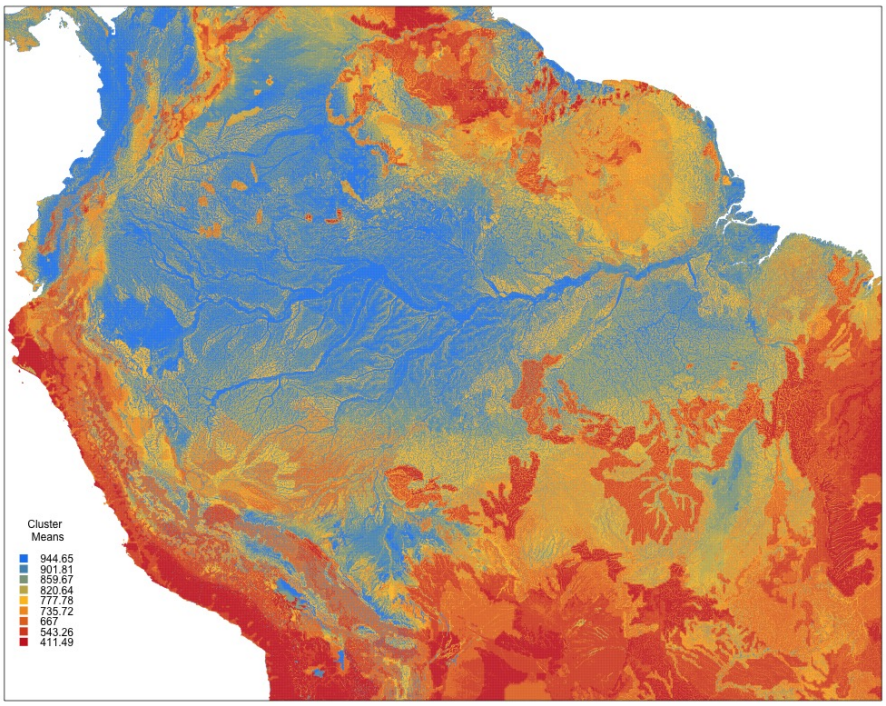


Figure 1: Map of study region with 9-means clustering of PCA of average soil moisture. This figure was generated using R statistics and graphics program using the "sp" package.

Using the R statistical programming software, a finite mixture of Bayesian linear regressions is estimated for soil moisture data in a region of Central and South America. Chapter Two introduces the methodology used in the analysis including Bayesian hierarchical models, finite mixture models, and grid search methods. Chapter Three describes the dataset and its preparation for analysis. In Chapter Four, the full hierarchical model is defined along with a sampling scheme. The results of the analysis are given in Chapter Five. Finally in Chapter Six, conclusions are drawn and further suggestions for modeling improvements are given.

Chapter Two: Methodology

Monte Carlo simulation

The fundamental concept of Bayesian analysis can be summarized in the belief that both observed and unobserved quantities arise from probability distributions. With all beliefs summarized in probability statements, a full probability model, $p(\theta|y)$ known as the joint posterior distribution, reflects a researchers understanding of the underlying behavior of all quantities and the data generation processes. Using Bayes theorem

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(y|\theta)p(\theta)}{p(y)},$$

where θ are the parameters of the model and y is the observed data, allows one to think conditionally. Here, $p(\theta)$ is the prior distribution of a parameter which reflects one's understanding of the model parameters. Because y is known, $p(y)$, commonly thought of as the normalizing constant which makes the distribution integrate to one, may be dropped as it remains constant and can be recovered with integration. By multiplying the likelihood $p(y|\theta)$ by the prior $p(\theta)$, we have the joint posterior $p(\theta|y)$ up to a constant of proportionality. Now, conditioned on all other known quantities, the distribution of unknown

parameters is known, regardless of whether or not it is of recognizable form.

As noted earlier, the selection of prior distributions sets Bayesian analysis apart from its frequentist counterpart. The foundation of Bayesian inference is the philosophy of subjectivity; a researcher may build in their prior beliefs about the distribution, dependence structures, and true value of a parameter when constructing a model. While not the case with all problems, the repetition of formulating priors gives a sort of conventional set of objective choices. However, it remains that the functional form of the data and the parametric form of the of distributions are up to the designer of the study. While priors are a way to quantify uncertainty in the true value of a parameter, because the likelihood is given, prior selection essentially controls the posterior obtained and the fit of the model.

Once the full joint posterior is obtained, the conditional posterior distributions can be easily gathered by marginalizing other quantities. Markov chain Monte Carlo (MCMC) simulation allows sampling from seemingly complex joint distributions that would otherwise involve complex integration by taking many samples from the parameter space. A Markov chain is a sequence of events whose distribution depends only on the previous event. Applied to sampling, draws are taken from

the conditional distributions given the previous draw of the other parameters and the data; together these specify a draw from the joint.

When the prior density and the sampling distribution are from the same family, the resulting conditional posterior will be in the form of a known distribution; these are called conditionally conjugate. With an MCMC method called Gibbs sampling, transition probabilities are defined by sampling directly from known distributions, which is simple with many statistical softwares. Otherwise, if the distribution is not conjugate, there exist many other algorithms for sampling from the joint, the most widely-known of which is the Metropolis-Hastings algorithm. MCMC works by drawing values of the parameters from their approximate distributions, the conditional posterior distributions. At each iteration, using the estimated values of the other parameters at the previous iteration, the draws are corrected to better approximate the target distribution. Because the approximate distributions are improved at each iteration, the draws eventually converge on the target distribution. Therefore, by iteratively sampling from each parameter's conditional distribution given the values of the other parameters, they build up a Monte Carlo sample from the joint posterior distribution of all model parameters. For a more detailed summary of MCMC see Gelman (2003).

Hierarchical Bayesian models

Much of the benefit of a Bayesian model comes from the ability to model complex dependence structures. Model structures with many levels, as frequently occur in real-life problems, must be modeled with those levels in mind. Non-hierarchical models with few parameters will poorly fit the data while models with many parameters tend to overfit the data and produce poor out-of-sample predictions (Gelman, et al., 2003, p. 117). Bayesian models, where parameters are treated as random, tend to preform well in circumstances that call for multi-level modeling.

Suppose the observed data y_{ij}, where i indexes units and j indexes groups, are drawn from some data generation process where each group's parameters θ_j are drawn from a population distribution of parameters. Further, assume the population distribution of parameters is governed by a set of hyperparameters ϕ. This hierarchical model may be expanded as far as necessary to reflect the natural structure of the data. In frequentist statistics, θ and ϕ are assumed fixed based on the data, however, to a Bayesian, the parameters are treated as random draws from distributions of parameters. From Gelman (2003), the simplest multi-level Bayesian model is given by:

1. Likelihood: $y_{ij}|\theta_j \sim p(y_{ij}|\theta_j)$

2. Prior: $\theta_j|\phi \sim p(\theta_j|\phi)$

3. Hyperprior: $\phi \sim p(\phi)$

Note that this model may be expanded as needed and parameters within the model may be fixed at any point; these choices are up to the researcher. Also important to notice is that the data y_{ij} is affected by ϕ only through the θ_j. Because of this, we may think locally about parameters of the model and use only the values which they are dependent on in their estimation.

Data augmentation of a finite mixture model

Suppose observations $y_1, \cdots, y_n$ are drawn from one of J categories, in the case of this study, distributions. For simplicity, the categories will be treated as finite and fixed, although algorithms exist for problems in which the number of components of the mixture are not known. The sampling model for a data point y_i is given by

$$f(y_i) = \sum_{j=1}^{J} w_j f_j(y_i), \quad i = 1, \cdots, n,$$

where the densities $f_j(y_i)$ are known up to some parameter $\theta_j = (\theta_1, \cdots, \theta_p)$ that includes the proportions, and the proportions are such that $(0 \leq w_j \leq 1)$ and $\sum_{j=1}^{J} w_j = 1$.

For posterior simulation, it is convenient to introduce latent indicators z_{ij} to match the data y_i with terms from the mixture (Diebold & Robert, 1994). For $\{i \in 1, \cdots, n\}$, z_i is a J-dimensional vector where $z_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{J} z_{ij} = 1$. Now the sampling model of the complete data (y_i, z_i) is given by

$$p(y_i, z_i|\mathbf{w} \cdots) = \prod_{j=1}^{J} \mathbb{1}(z_{ij} = 1) w_j f_j(x_i)$$

This sets up the full hierarchical model with the data $x \sim f(y|\theta, z)$ at the base of the hierarchy, followed by the latent indicators $z \sim f(z|\theta)$, and finally the distributions of the true parameters on top.

Now to sampling from this model is relatively simple. Suppose we have n latent indicators which will each have a success in exactly one of J categories; this describes the multinomial distribution, a generalization of the binomial distribution, where each category has a fixed success probability (Albert & Chib, 1993). At each iteration of the MCMC the indicators z_i are sampled using the posterior

probability with which they are allocated to each component

$$p_j = \frac{w_j f_j(y_i)}{\sum_{j=1}^{J} w_j f_j(y_i)},$$

using the current values of the parameters within the MCMC simulation (Viele, 2002, p. 319). Conditioned on values for the latent indicators $z_i = j$, the distribution of the response data is given by $f(y, \theta_j | z = j)$. In other words, distinct component parameters may now be estimated using only the data from the component from which they are drawn. The parameters within a given component are now updated with a Gibbs step (when available) or another method such as the Metropolis-Hastings algorithm.

Grid approximation

Although the Metropolis-Hastings algorithm provides a relatively simple way to sample from the posterior without explicitly deriving the normalizing constant, the chain moves based on an acceptance probability which subjects simulation to potential problems related to slow mixing or simulations getting trapped in local extrema (Brooks, 2011). A discrete approximation allows for direct sampling similar to the Gibbs sampler. Another appealing feature of the grid

approximation is the ability to sample multi-dimensional parameters simultaneously. To obtain the unnormalized posterior, simply multiply the likelihood and the non-conjugate prior. However, the unnormalized posterior is not a probability density function because it does not integrate to 1. The normalizing constant is simply the area under the unnormalized posterior. Clearly, dividing the unnormalized density by this constant now defines a probability function. Calculating this constant involves integrating over the domain of the density, which is difficult in many cases. However, grid approximation offers an easy approximation to the normalizing constant. The algorithm for grid approximation of the posterior distribution of a set of parameters θ is given in Hoff (2006) as

1. Define a discrete grid of points $\{\theta_1, \theta_2, \cdots, \theta_K\}$ in the domain of $p(\theta|\mathbf{y})$ where we expect to observe the parameter.

2. Compute $p(\theta_k)p(\mathbf{y}|\theta_k)$ for each point in the grid.

3. Compute the approximate normalizing constant $c \approx \sum_{k=1}^{K} p(\theta_k)p(\mathbf{y}|\theta_k)$.

Now

$$\tilde{p}(\theta|\mathbf{y}) = \frac{p(\theta)p(\mathbf{y}|\theta)}{\sum_{k=1}^{K} p(\theta_k)p(\mathbf{y}|\theta_k)} = \frac{\tilde{p}(\theta)p(\mathbf{y}|\theta)}{\tilde{p}(\mathbf{y})}$$

is a discrete approximation to the posterior where $\tilde{p}(\theta)$ is a discrete approximation of the prior and $\tilde{p}(\mathbf{y})$ is the marginal distribution of $\mathbf{y}$ based on the discrete prior. Therefore we can sample from this density using this set of probabilities. Although simple to set up, this method does have some drawbacks. First, if the choice of the grid does not cover the proper range, the samples will be poor approximations to the posterior. Secondly, the approximation is discrete; the grid must be made very fine to simulate random draws from a continuous distribution. Still, grid approximation offers an easy alternative to other sampling methods when the posterior is non-conjugate.

Chapter Three: Data

The data consists of a simulation of soil moisture from the Miguez-Macho and Fan 2012 study "The role of groundwater in the Amazon water cycle." For the years 2000-2011, both soil moisture and water table depth are simulated at four minute intervals on a fine grid over the northern region of South America (terrestrial portion bounded by 9.13°N, 81.50°W and 20.50°S, 44.01°W). This 2250×1780 grid consists of approximately 3.3 million terrestrial cells at a resolution of 1 arc-minute (~2km) per cell. The data is produced using their LEAF Hydro-flood model which they cross-validate using observed soil-moisture and evapotranspiration data. Soil moisture measures are the volumetric water content in the top two meters of soil and range from 146mm to 984mm.

Chapter Four: Model and Implementation

While there are more complex time-series models, such as seasonal ARIMA processes, that may explain more variation within the data, the primary focus of this study is the seasonality of climate patterns. The simplest method to approach this problem is to use monthly means from the eleven years of data at a given pixel rather than the entire dataset. Let y_i denote the twelve-element vector of monthly averages at a given pixel. Note, it is not possible for monthly observations within the same pixel to be in different clusters. Next, set up the design matrix with an intercept and dummy variables for all other months of observations. Simple linear regression will provide coefficients for monthly means within a cluster. Analytically, the estimates for the monthly coefficients obtained from the full dataset versus those from the dimension-reduced data should be the same, however, the computation time is dramatically reduced.

We will work with a simple two-level hierarchical Bayesian linear regression model of the data y_{ij} with cluster level coefficients. Denote the likelihood of each observation, the vector of average monthly values

at a particular pixel i, as

$$\mathcal{L} = p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{w}, \boldsymbol{\sigma}^2) = \prod_{i=1}^{n} \sum_{j=1}^{J} w_j \; N(y_{ij}|\mathbf{X}_{ij}\boldsymbol{\beta}_j, \sigma_j^2\mathbf{I})$$

where j is an indicator of the cluster of the pixel. Alternatively, using latent indicators for each observation vector z_i as was demonstrated in Chapter Two, an identical model can be written as

$$p(y_i|\boldsymbol{\beta}, \sigma^2, z_i = j) = N(y_i|\mathbf{X}_j\boldsymbol{\beta}_j, \sigma_j^2\mathbf{I}) \cdot \mathbb{1}(z_i = j)$$
$$p(z_i = j) = w_j$$

where β_j is the regression coefficient vector for predicting average monthly soil moisture in a given cluster and σ_j^2 is the data-level variance. Now, the clusters must be defined because the points within a given cluster may change at each iteration of the MCMC. Let the first cluster $j = 1$ be given as the cluster with the smallest estimated intercept and progress sequentially to define the remaining clusters (Jasra, 2005, p. 53). Level two of the hierarchy is defined as

$$p(\beta_j|\mu_{\beta_j}, \tau_j^2) = N(\beta_j|\mu_{\beta_j}, \tau_j^2 I), \quad j = 1, \cdots, J,$$

where μ_{β_j} is the mean vector from which coefficients are drawn and τ_j^2 is the coefficient-level variance. Because each coefficient is measured on

the same scale, assuming equal variance of the coefficients does not impose unreasonable restrictions.

Prior Selection

Due to the large amount of data available, selecting "non-informative" priors allows for asymptotics to dominate the estimation of the distributions of parameters. With large sample sizes, the data is assumed to be a sufficient representation of the true population. As such, priors for the data-level parameters are uniform, or flat, expressing the belief that equal probability can be given to all areas of the parameter space. Following the recommendations presented in Gelman (2006), the a half-normal prior on τ^2 is selected because of its good behavior around zero and because it is proper. Additional information about the selection of priors for variance parameters can be found in Polson & Scott (2012). The full hierarchical model is completed with the set of priors

$$p(\mu_{\beta_j}, \sigma_j^2) \propto 1$$

$$p(\tau_j^2) \sim N^+(0, 10^2)$$

While the full joint posterior distribution is not conjugate, many of the parameters are conditionally conjugate. Figure 2 is a directed

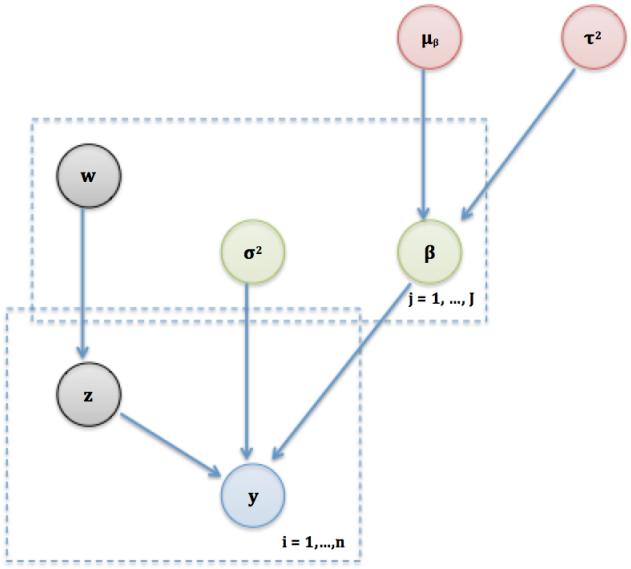


Figure 2: Directed acyclical graph of finite mixture of regressions

acyclical graph (DAG) of the model.

A moralized graph is a graph where the parents of common children are connected and all edges are made undirected. By moralizing the DAG, conditional dependency can be observed easily. The conditional distributions are as follows:

$$p(z_i|\mathbf{w}) \sim Multi(n, \mathbf{p})$$
$$\text{where } p_j \propto \frac{w_j N(y_{ij}|\boldsymbol{\mu_j}, \sigma_j^2)}{\sum_{j=1}^{J} w_j N(y_{ij}|\boldsymbol{\mu_j}, \sigma_j^2)}$$

$$p(\mathbf{w}|\mathbf{z}) \sim Dir(\alpha + n_1, \alpha + n_2, \cdots, \alpha + n_J)$$

$$p(\boldsymbol{\beta_j}|\sigma_j^2, \tau_j^2, \boldsymbol{\mu}_{\beta_j}) \sim N(m, V)$$
$$\text{where } V = [\mathbf{X}^T(\sigma_j^{-2} I_n)\mathbf{X} + \tau_j^{-2} I_p]^{-1}$$
$$\text{and } m = V[\mathbf{X}^T(\sigma_j^{-2} I_n) + \boldsymbol{\mu}_{\boldsymbol{\beta_j}}(\tau_j^{-2} I_p)]$$

$$p(\sigma^{-2}|\boldsymbol{\beta_j}) \sim Ga(\frac{n_j}{2}, (\mathbf{y_j} - \mathbf{X_j}\boldsymbol{\beta_j})^T(\mathbf{y_j} - \mathbf{X_j}\boldsymbol{\beta_j}))$$

$$p(\boldsymbol{\mu}_{\beta_j}|\boldsymbol{\beta_j}, \tau_j^2) \sim N(\boldsymbol{\beta_j}, \tau^2)$$

$$p(\tau^2|\boldsymbol{\beta_j}, \boldsymbol{\mu}_{\beta_j}) = N(\boldsymbol{\beta_j}|\boldsymbol{\mu}_{\beta_j}, \tau_j^2) \cdot N^+(\tau^2|0, 10^2)$$

Notice that all but one of the these distributions are have conditionally conjugate posteriors. For parameters with conditionally conjugate distributions we implement transition probabilities to update those parameters with Gibbs sampling. For sampling from the conditional posterior of τ^2, a few options exist. The traditional approach is to use a Metropolis-Hastings algorithm, however, another option that may be simpler and potentially more efficient is the grid search, described in Chapter Two.

Using a random sample of 2,500 pixels from the study region, an

MCMC is run for 5,000 iterations. The R statistical programming software is used, and code is provided in the appendix. With the resulting estimates of the monthly regression coefficients, predicted values for each month within a given cluster are constructed. In order to cluster the remainder of pixels in the study region, a distance matrix is created representing the Euclidean distance between the monthly observations at a pixel and each of the estimated monthly values within clusters. Pixels are then assigned to the cluster with the minimum Euclidean distance.

Chapter Five: Results and Discussion

A convergent MCMC provides a full probabilistic description of all uncertainties about a model for the data. With uncertainty implicit in parameter estimation, many argue that the predictive ability of the model is more realistic. An estimation of average monthly soil moisture using a finite mixture of multivariate normal densities offers a better understanding of the periodicity of soil moisture in the Amazon region. This section provides a brief overview of results and model diagnostics.

The first, and arguably most important, feature to check is convergence of the Markov chains. If the chains have not mixed well (explored the parameter space efficiently), the model does not offer a valid representation of the distributions of the parameters (Lam, 2009b). Trace plots, autocorrelation measures, and density estimates are provided in the Appendix. While numerical methods for assessing convergence exist, a visual inspection provides evidence that the parameters have stabilized after approximately 400 iterations. These first iterations are often known as the burn-in period. After the burn-in, a well-mixing model should provide independent draws from the target distribution. An autocorrelation plot provides the correlation between every draw and its lags; low correlation indicates

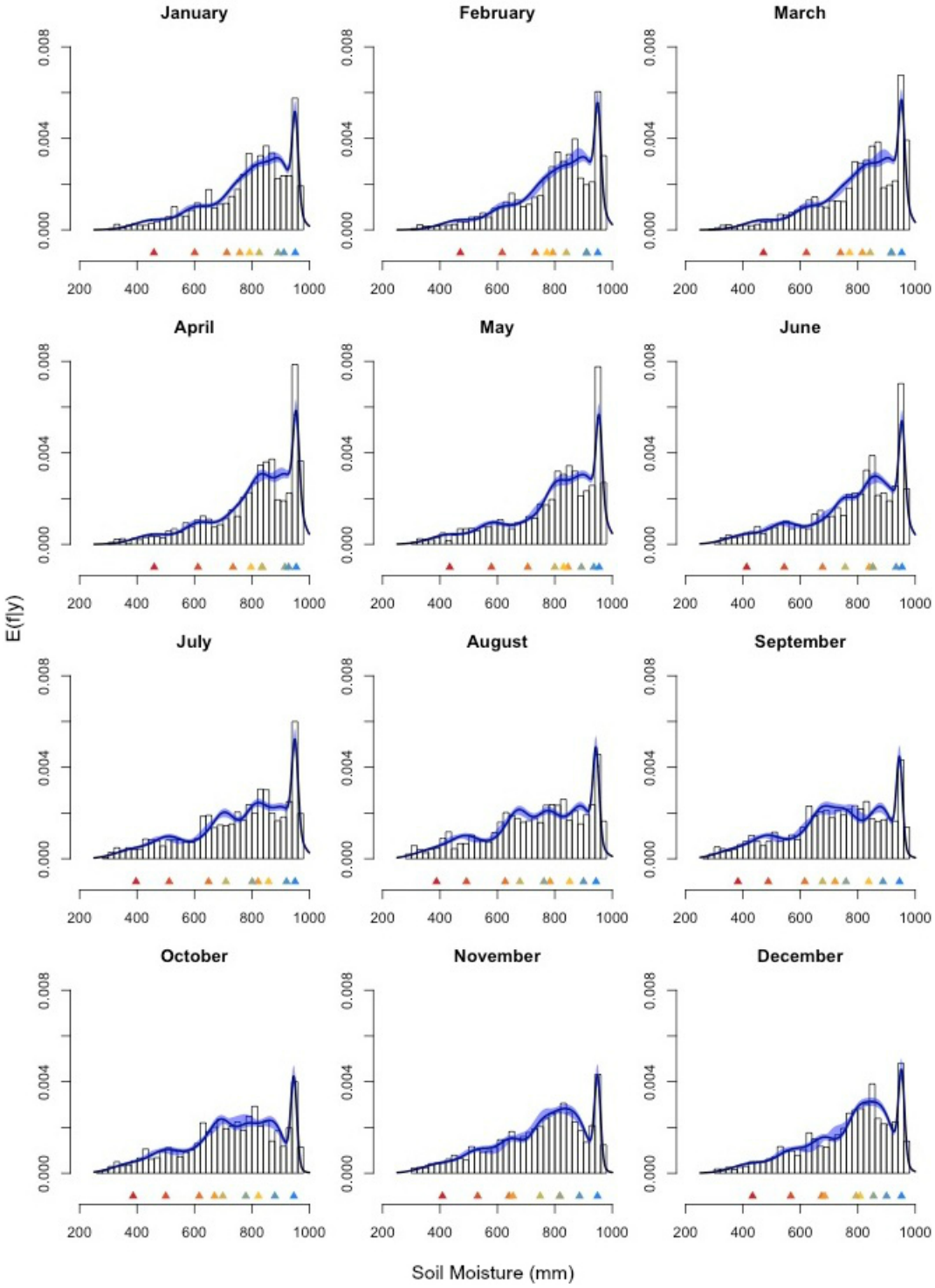


Figure 3: Posterior density estimate at each month with 95% point-wise interval estimates and predicted cluster means shown by arrows. This figure was generated using R statistics and graphics program.

independent draws. An optimal autocorrelation is evident in the draws for group standard deviation σ and many of the weights. In contrast, the draws for the parameter standard deviation τ and the monthly betas are correlated over more lags. Thinning is a method used to obtain independent samples by taking every k^{th} iterate instead of every sample. The thinned samples of beta and tau should be more representative of independent draws from their posteriors.

Next, the fit of the model may be evaluated. Figure 3 shows the estimated density for each month along with the predicted means for each cluster. In most months, the predicted density shown by the line closely follows the observed density shown with a histogram. While potentially difficult to implement in a mixture model, one way to improve this fit may be to allow different monthly observations at a particular pixel to be placed in different clusters. The theory behind this idea is that different soil types or topographies may be

	w[1]	w[2]	w[3]	w[4]	w[5]	w[6]	w[7]	w[8]	w[9]
w.predicted	0.072	0.107	0.124	0.055	0.137	0.123	0.129	0.157	0.098
w.fullmap	0.066	0.112	0.117	0.061	0.114	0.133	0.128	0.135	0.134

Table 1: Weight estimates for each cluster from the MCMC compared to the observed proportions when the cluster with the highest likelihood is assigned to each pixel in the full map

exchangeable between clusters. Overall, however, the fit of the predicted density at each month is very good.

With draws that are assumed to be independent samples from the target distribution, the density may be estimated at all values of soil moisture within the range of the data. While particular pixels do not define a cluster, having a full probability model provides the ability to evaluate the likelihood that a particular pixel is drawn from a cluster. Table 1 shows the predicted proportions of pixels within each cluster

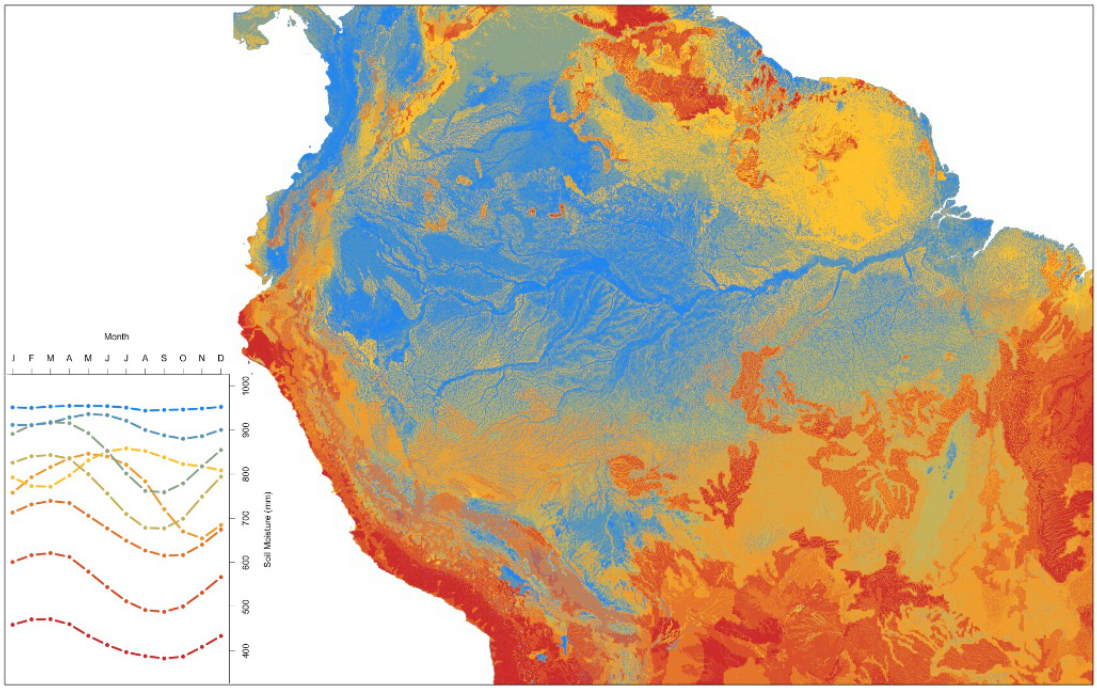


Figure 4: Map of study region with clustering from a 9-group finite mixture model. This figure was generated using R statistics and graphics program using the "sp" package.

compared to the total proportion of pixels when projected onto the map using the maximum likelihood cluster estimate. Figure 4 provides a map in which pixels are assigned to the cluster with the greatest likelihood. While the coloration in the map does not appear to be widely discrepant, in fact, 40.9% of pixels were put in a different cluster than those specified in the preliminary analysis as shown in Figure 5. This implies that clustering on overall average soil moisture will not provide consistent estimates of the periodic nature of the data, and thus the average periodic behavior of a cluster is not consistent using

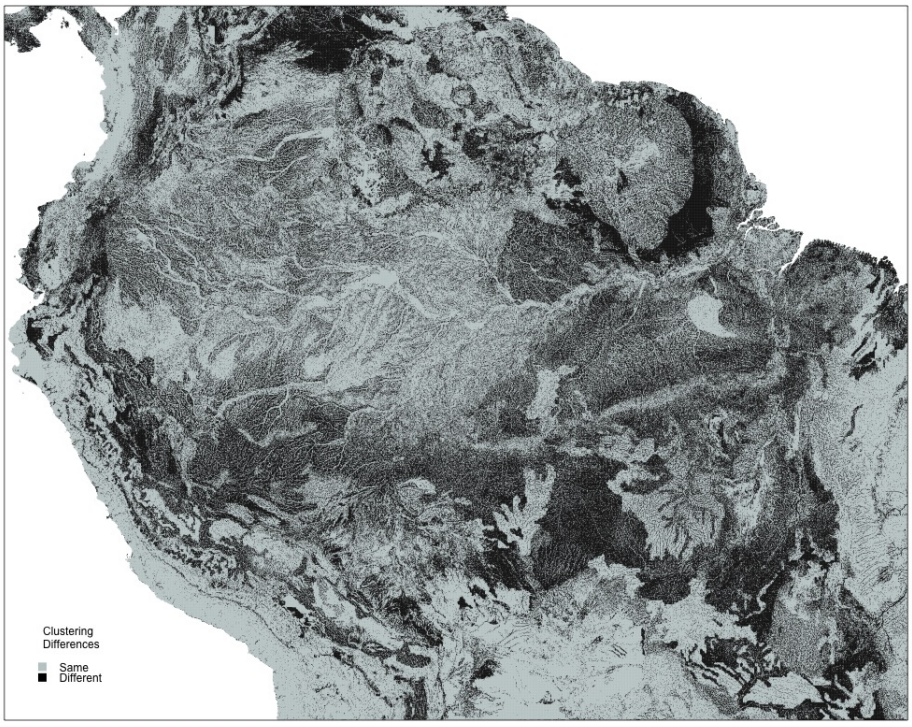


Figure 5: Map of study region showing differences in clustering between the preliminary analysis and the mixture model predictions. This figure was generated using R statistics and graphics program using the "sp" package.

PCA and k-means.

While this study was performed primarily as a demonstration of methods, it offers predictions which can be used in a variety of inferences. In the case of soil moisture, especially in the Amazon region, there are ecological, geographical, and environmental consequences to slight changes in water content, the reasons for which may include deforestation or pollution. The periodicity represented in the clusters can be taken as an accurate representation of the soil moisture behavior over time for a given region, soil type, or other aggregate. This model provides a probabilistic base for which to evaluate similar periodic or seasonal changes in other studies.

Chapter Six: Conclusion

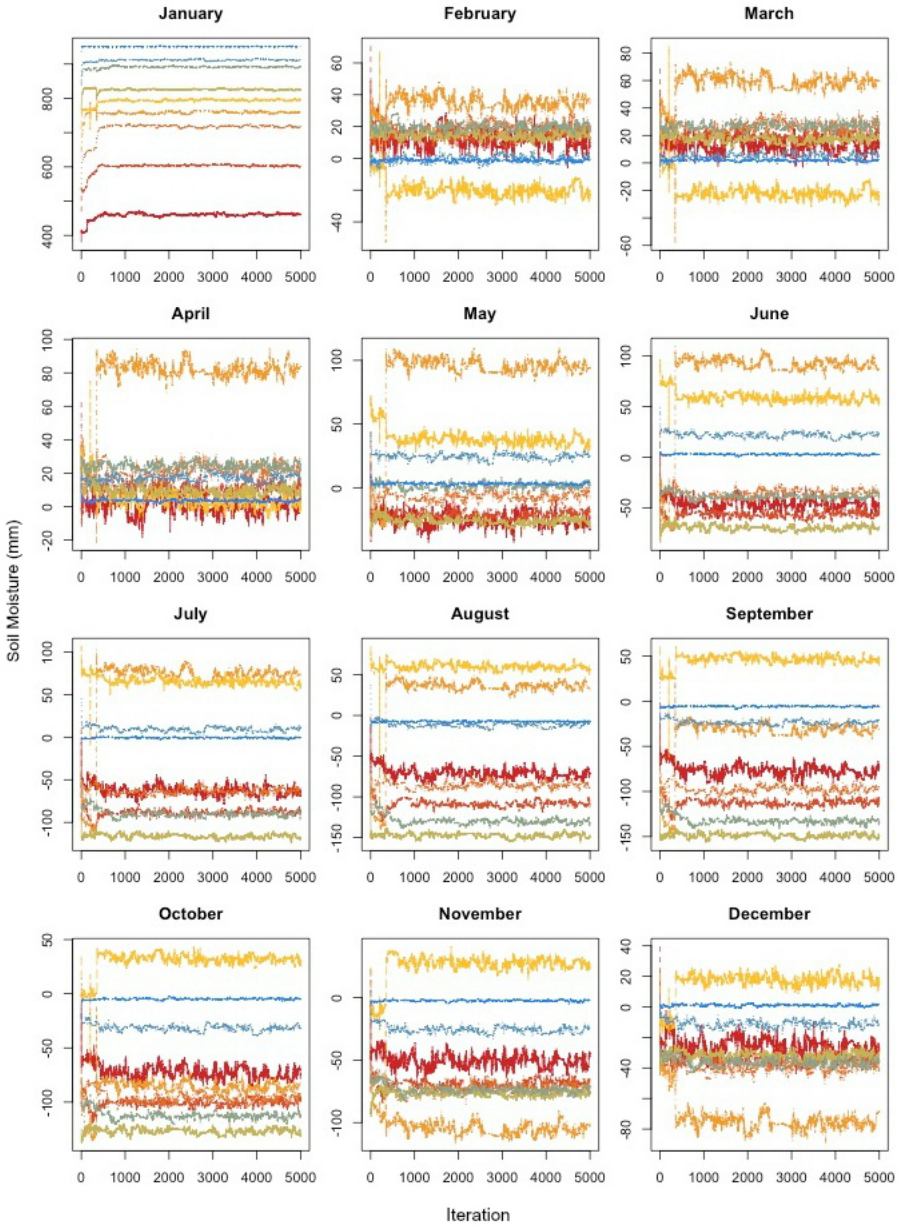
The Bayesian finite mixture model offers an effective alternative to k-means cluster estimation which can be constructed to model specific characteristics determined by the researcher. While this model of soil moisture offers different predictions than a simple clustering of overall averages, there are still improvements that could be made. Nine was the number of clusters chosen because the Zaramea study suggested this was consistent with the literature on climatic regions in the study area in South America. Alternatively, there are Bayesian algorithms, such as reversible jump, that allow for a change in the dimensions of the parameter space to sample the number of clusters in addition to all other parameters. Another potential improvement is the introduction of spatial dependence among pixels either through spatial smoothing or through the construction of the covariance matrices. Finally, given the complexity of the simulation model for which the data was created, repeating this study using other inputs could provide additional insights to the true relationship of various environmental processes.

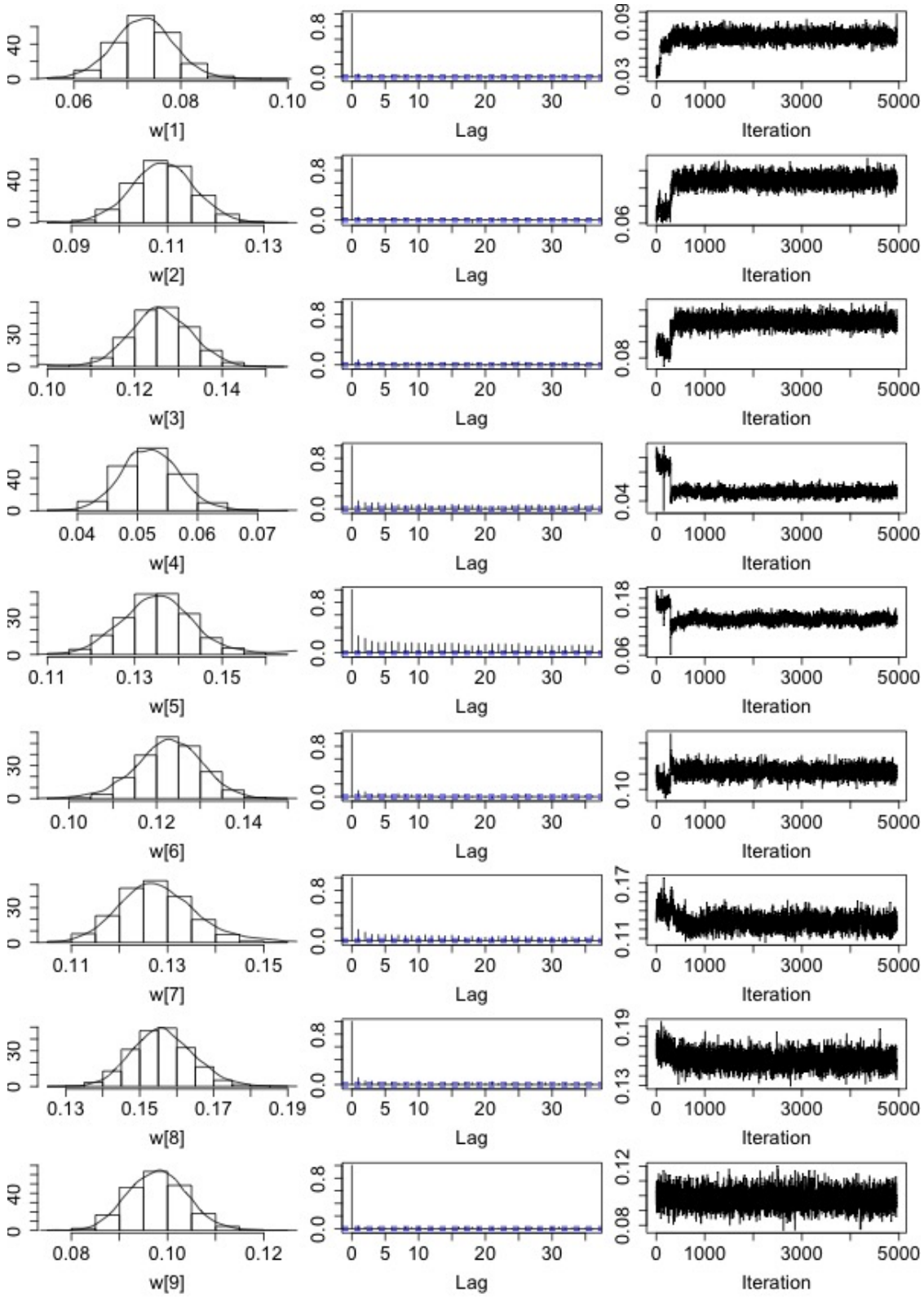
In summary, the proposed inference provides a coherent, model-based description of all uncertainties. Compared to purely empirical estimates, such descriptions have the advantage of being

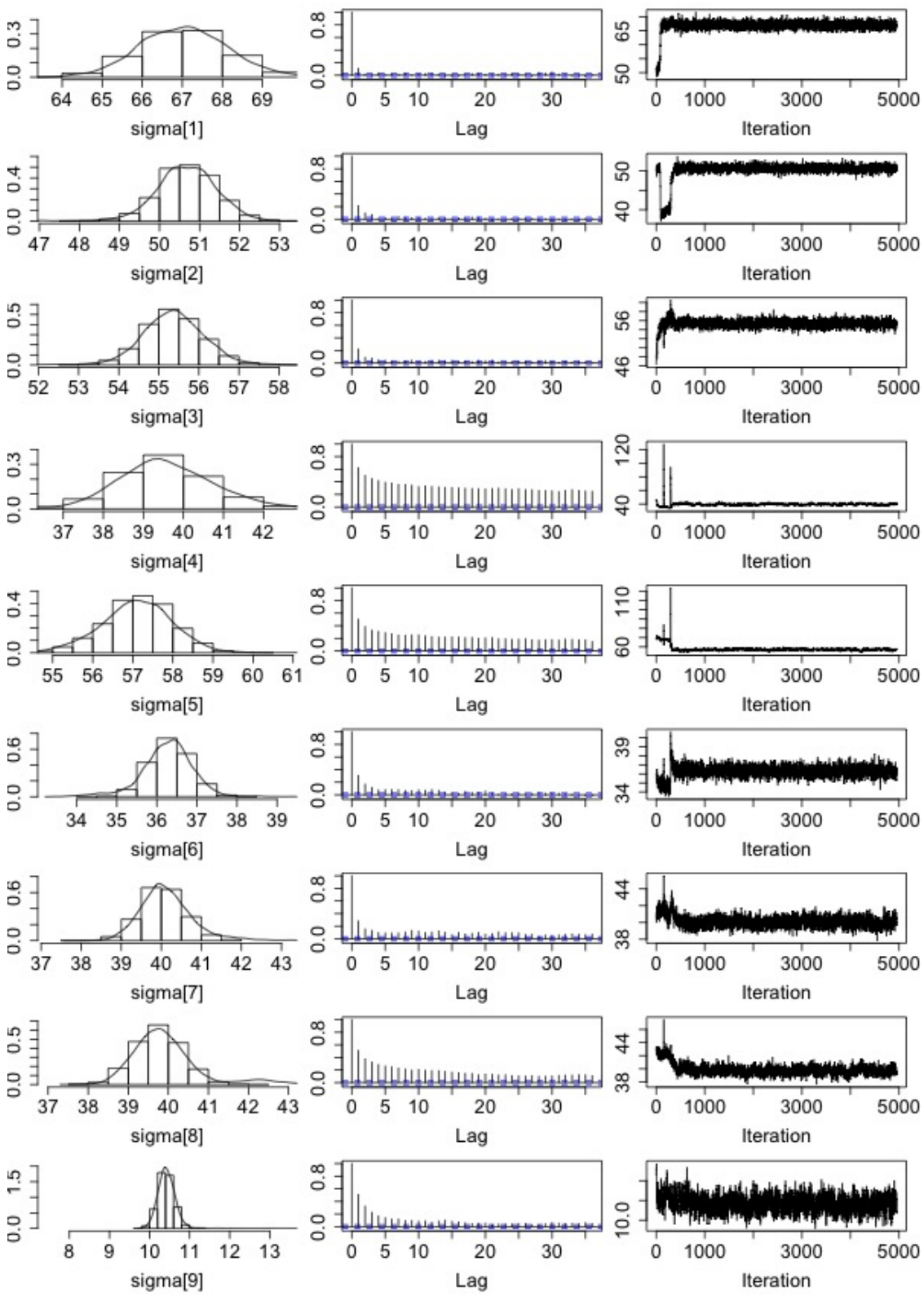
consistent and logically coherent across any number of tests, estimates, time frames, etc. Point-based estimates seem deficient if they do not also capture how certain that estimate is. In particular, a description of uncertainties is important to related decision problems. Examples are decisions about the placement of monitoring stations or policy decisions about development. The Bayesian finite mixture model offers a detailed representation of both the observed data as well as the its generation process which may be used to provide sound, probabilistic predictions in many related problems.

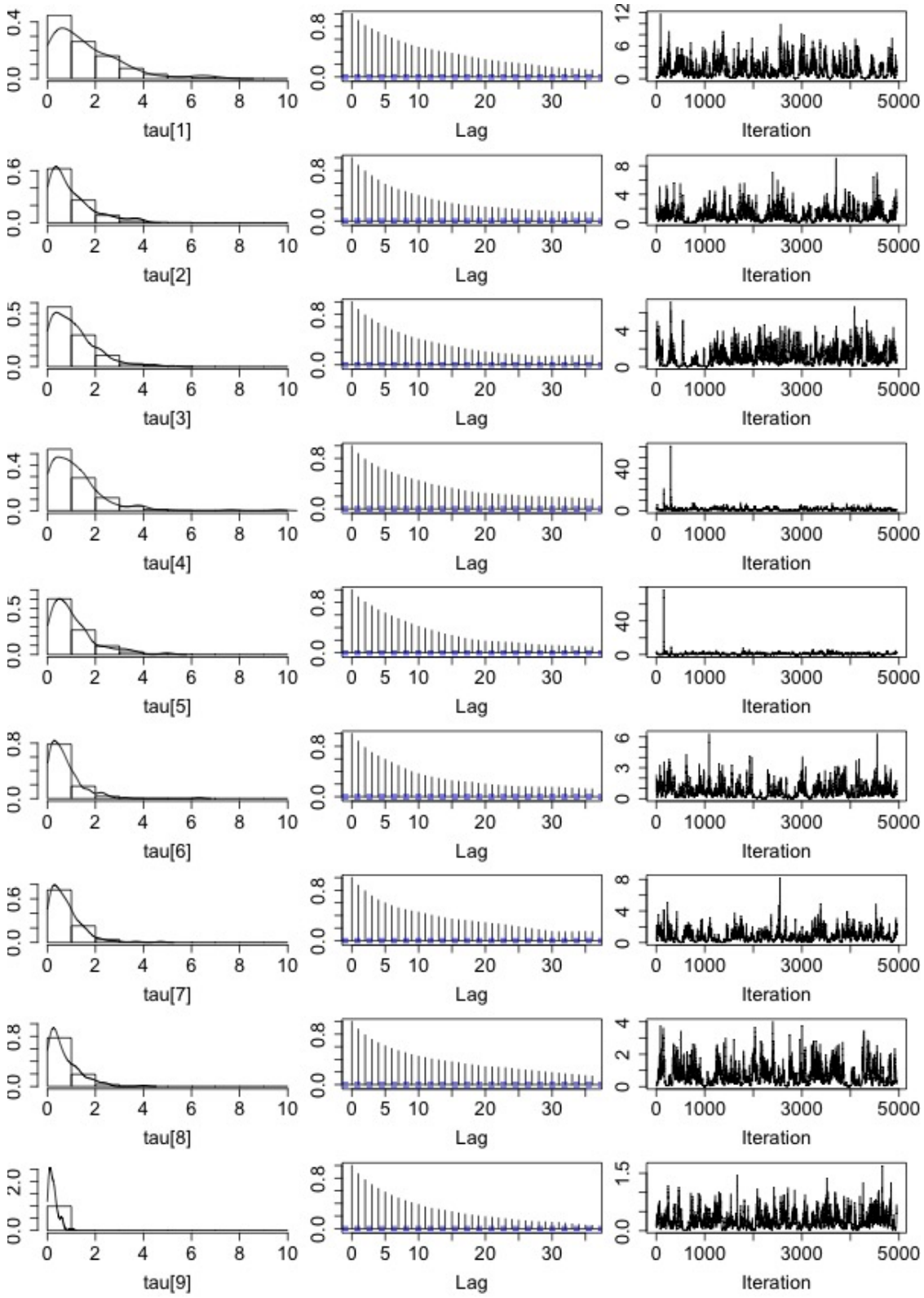
Appendix

Diagnostic plots:




w[1]
w[2]
w[3]
w[4]
w[5]
w[6]
w[7]
w[8]
w[9]
Lag
Iteration


sigma[1]
sigma[2]
sigma[3]
sigma[4]
sigma[5]
sigma[6]
sigma[7]
sigma[8]
sigma[9]
Lag
Iteration


tau[1]
tau[2]
tau[3]
tau[4]
tau[5]
tau[6]
tau[7]
tau[8]
tau[9]
Lag
Iteration

R code:

```
bayes.mixreg <- function(y,x,J,n.iter=1000,xgrid){
  require(gtools) ; require(mvtnorm) ; require(truncnorm) ; require(car)
  y <- as.vector(y)
  x <- as.matrix(x)

  # sample z ~ p(z|y,beta,sigmasq,w)
  sample.z <- function(y,x,beta,sigmasq,w){
    n <- length(y)/12
    z <- rep(0,n)
    for(i in 1:n){
      yi <- y[(12*i-11):(12*i)]
      pr <- rep(0,J)
      for(j in 1:J){
        pr[j] <- w[j]*dmvnorm(yi,x[1:12,]%*%beta[,j],sigmasq[j]*diag(12))
      }
      pr <- pr/sum(pr)
      z[i] <- sample(1:J,1,replace=T,prob=pr)
    }
    return(z)
  }

  # sample w ~ p(w|z)
  sample.w <- function(z){
    a <- rep(0,J)
    for(j in 1:J){
      Aj <- which(z==j)
```

```
        nj <- length(Aj)
        a[j] <- alpha + nj
    }
    w <- rdirichlet(1,a)
    return(as.vector(w))
}

# sample beta ~ p(beta/y,sigmasq,mu,tau)
sample.beta <- function(y,x,sigmasq,tau,mu){
    n <- length(y) ; p <- ncol(x)
    V <- solve(((1/tau^2)*diag(p)) + t(x)%*%((1/sigmasq)*diag(n))%*%x)
    m <- V%*%(t(x)%*%((1/sigmasq)*diag(n))%*%y + mu/tau^2)
    return(rmvnorm(1,m,V))
}

# sample mu ~ p(mu/beta,tau)
sample.mu <- function(beta,tau){
    p <- length(beta)
    V <- (tau^2/p)*diag(p)
    m <- beta
    return(rmvnorm(1,m,V))
}

# sample sigmasq ~ p(sigmasq/y,beta)
sample.sigmasq <- function(y,x,beta){
    n <- length(y)
    return(1/rgamma(1,n/2,t(y-x%*%beta)%*%(y-x%*%beta)/2))
}
```

```
# evalutate the posterior density of a tau
posterior.tau <- function(beta,mu,tau,s){
    log.like <- sum(dnorm(beta,mu,sd=tau,log=T))
    log.prior <- log(dtruncnorm(tau,a=0,sd=s))
    log.post <- log.like + log.prior
    return(exp(log.post))
}

# sample tau using grid approximation of p(z|beta,mu)
sample.tau <- function(beta,mu,s){
    grid <- seq(0.0001,s,length=10000)
    pr <- rep(0,length(grid))
    for(p in 1:length(grid)){
        pr[p] <- posterior.tau(beta,mu,grid[p],s)
    }
    return(sample(grid, 1, prob = pr))
}

# enforce clusters with ascending beta_0
B0 <- seq(1,length(y),12)
Recode <- function(y,z.long){
    df <- data.frame(y,z.long)
    clus <- aggregate(y~z.long,data=df[B0,],mean)
    clus <- clus[order(clus$y),] ; clus <- cbind(clus,o = 1:J)
    clus <- clus[order(clus$z.long),] ; clus$z.long <- letters[clus$z.long]
    z.long <- letters[z.long]
    return(recode(z.long,paste(paste0(" ' ",clus$z.long,"'="",clus$o),collapse=';')))
}
```

```
# evaluate joint posterior density at all points on xgrid
f <- function(xgrid, w,beta,sigmasq){
   ymat <- matrix(0,nrow=length(xgrid),ncol=12)
   for(i in 1:length(xgrid)){
      y <- rep(0,12)
      xi <- rep(xgrid[i],12)
      for(j in 1:J){
         y <- y + w[j]*dnorm(xi,x[1:12,]%*%beta[,j],sd=sqrt(sigmasq[j]))
      }
      ymat[i,] <- y
   }
   return(ymat)
}

# Hyperparameters
alpha <- 1 ; s <- 100

# Initialize parameters
sigmasq <- rep(1000,J) ; tau <- rep(1,J)
ymat <- matrix(y,nrow=sample.size,ncol=12,byrow=T)
d <- dist(ymat)
hc <- hclust(d)
z <- cutree(hc,J)
zmat <- matrix(z,nrow=12,ncol=length(z),byrow=T)
z.long <- as.vector(zmat)
z.long <- Recode(y,z.long)
a <- rep(0,J)
for(j in 1:J){
```

```
    Aj <- which(z==j)
    nj <- length(Aj)
    a[j] <- alpha + nj
}
w <- as.vector(rdirichlet(1,a))
beta <- matrix(0,nrow=12,ncol=J)
mu.beta <- matrix(0,nrow=12,ncol=J)
for(j in 1:J){
    yj <- as.vector(y[z.long==j])
    xj <- as.matrix(x[z.long==j,])
    mu.beta[,j] <- t(xj)%*%yj/length(yj)
    beta[,j] <- as.vector(sample.beta(yj,xj,sigmasq[j],tau[j],mu.beta[,j]))
    tau[j] <- sample.tau(beta[,j],mu.beta[,j],s)
}

# create bookkeeping objects
f.array <- array(dim=c(length(xgrid),12,n.iter))
beta.array <- array(dim=c(12,J,n.iter))
w.mat <- matrix(0,ncol=J,nrow=n.iter)
tau.mat <- matrix(0,ncol=J,nrow=n.iter)
sigma.mat <- matrix(0,ncol=J,nrow=n.iter)

# run the sampler for n.iter iterations
for(it in 1:n.iter){
    z <- sample.z(y,x,beta,sigmasq,w)
    zmat <- matrix(z,nrow=12,ncol=length(z),byrow=T)
    z.long <- as.vector(zmat)
    z.long <- Recode(y,z.long)
```

```
        print(table(z.long)/12)
        w <- sample.w(z)
        for(j in 1:J){
            yj <- y[z.long==j]
            xj <- x[z.long==j,]
            mu.beta[,j] <- sample.mu(beta[,j],tau[j])
            beta[,j] <- sample.beta(yj,xj,sigmasq[j],tau[j],mu.beta[,j])
            sigmasq[j] <- sample.sigmasq(yj,xj,beta[,j])
            tau[j] <- sample.tau(beta[,j],mu.beta[,j],s)
        }
        print(it)
        # store values from each iteration
        f.array[,,it] <- f(xgrid,w,beta,sigmasq)
        beta.array[,,it] <- beta
        w.mat[it,] <- w
        tau.mat[it,] <- tau
        sigma.mat[it,] <- sqrt(sigmasq)
    }
    chains <- list(f=f.array,beta=beta.array,w=w.mat,tau=tau.mat,sigma=sigma.mat)
    return(chains)
}
y <- dat.monthly$mean.value
x <- model.matrix(mean.value~month,dat.monthly)
colnames(x)[2:12] <- month.abb[2:12]
n.iter <- 5000
J <- 9
xgrid <- seq(251,1000,1)
g <- bayes.mixreg(y,x,J,n.iter,xgrid)
```

References

Albert, J. H., & Chib, S. (1993). Bayesian analysis of binary and polychromous response data. *Journal of the American Statistical Association, 88*(422), 669-679.

Brooks, S., Gelman, A., Jones, G. L., & Meng, X. L. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC.

Diebolt, J., & Robert, C. P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *Journal of the Royal Statistical Society, 56*(2), 363-375.

Dowle, M., Short, T., Lianoglou, S., Srinivasan, A., Saporta, R., & Antonyan, E. (2014). data.table: Extension of data.frame [Computer software manual]. Retrieved from http://CRAN.R-project.org/package=data.table (R package version 1.9.4).

Escobar, M. D., & West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association, 90*(430), 577-588.

Fox, J., & Weisberg, S. (2011). car: An R companion to applied regression (2nd ed.) [Computer software manual]. Thousand Oaks, CA: Sage. Retrieved from http://socserv.socsci.mcmaster.ca/jfox/Books/Companion (R package version 2.0-25).

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis, 1*(3), 515-533.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Texts in Statistical Science: Bayesian Data Analysis* (2nd ed.). New York, NY: Chapman & Hall/CRC.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., & Hothorn, T. (2014). mvtnorm: Multivariate normal and t distributions [Computer software manual]. Retrieved from http://CRAN.R-project.org/package=mvtnorm (R package version 1.0-2).

Hoff, P. D. (2006). *Introduction to Bayesian statistics for the social sciences.*

Hurn, M., Justel, A., & Robert, C. P. (2003). Estimation mixtures of regressions. *Journal of Computational and Graphical Statistics, 12*(1), 55-79.

Jasra, A., Holmes, C. C., & Stephens, D. A. (2005). Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science, 20*(1), 50-67.

Kass, R. E., & Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association, 91*(435), 1343-1369.

Keitt, T. (2013). rgdal2: Wraps the Geospatial Data Abstraction Library [Computer Software Manual]. (R package version 0.1).

Lam, P. (Presenter). (2009). *Convergence diagnostics.* Lecture presented at Government 2002: Bayesian Statistics in Political Science Research, Harvard University.

Lam, P. (Presenter). (2009). *Non-conjugate models and grid approximation.* Lecture presented at Government 2002: Bayesian Statistics in Political Science Research, Harvard University.

Lambert, P. C., Sutton, A. J., Burton, P. R., Abrams, K. R., & Jones, D. R. (2005). How vague is vague? A simulation study of the impact of the use of vague prior distributions in MCMC using WinBUGS. *Statistics in Medicine, 24*, 2401-2428.

Miguez-Macho, G., & Fan, Y. (2012). The old of groundwater in the Amazon water cycle: 2. Influence on seasonal soil moisture and evapotranspiration. *Journal of Geophysical Research, 117*(D15114), 1-27.

Müllner, D. (2013). fastcluster: Fast hierarchical, agglomerative clustering routines for R and Python [Computer software manual]. *Journal of Statistical Software, 53*(9), 1-18. Retrieved from http://www.jstatsoft.org/v53/i09/ (R package version 1.1.16).

Nychka, D., Furrer, R., & Sain, S. (2015). fields: Tools for spatial data [Computer software manual]. Retrieved from http://CRAN.R-project.org/package=fields (R package version 8.2-1).

Pebesma, E.J., Bivand, R.S. (2005). sp: Classes and methods for spatial data in R [Computer software manual]. Retrieved from http://cran.r-project.org/doc/Rnews/ (R package version 1.0-17).

Polson, N. G., & Scott, J. G. (2012). On the half-Cauchy prior for a global scale parameter. *Bayesian Analysis, 7*(4), 887-902.

R Core Team (2014). R: A language and environment for statistical computing [Computer software manual]. R Foundation for Statistical Computing, Vienna, Austria. Retrieved from http://www.R-project.org/.

Rai, P. (Presenter). (2011, October 4). *Data clustering: k-means and hierarchical clustering*. Lecture presented at CS5350/6350, University of Utah.

Trautmann, H., Steuer, D., Mersmann, O., & Bornkamp, B. (2014). truncnorm: Truncated normal distribution [Computer software manual]. Retrieved from http://CRAN.R-project.org/package=truncnorm (R package version 1.0-7).

Viele, K., & Tong, B. (2002). Modeling with mixtures of linear regressions. *Statistics and Computing, 12*, 315-330.

Warnes, G. R., Bolker, B., & Lumley, T. (2014). gtools: Various R programming tools [Computer software manual]. Retrieved from http://CRAN.R-project.org/package=gtools (R package version 3.4.1).

Zalamea, P. C., Munoz, F., Stevenson, P. R., Paine, C.E. T., Sarmiento, C., Sabatier, D., & Heuret, P. (2011). Continental-scale patterns of *Cecropia* reproductive phenology: evidence from herbarium specimens. *Proceedings of the Royal Society B, 278*, 2437-2445.

Vita

Thomas Edwin Cuthbertson was born in Birmingham, Alabama. After completing his work at John Carroll Catholic High School, Birmingham, Alabama in 2009, he entered Furman University in Greenville, South Carolina. In 2013, he received the degrees of Bachelor of Arts in Economics and Bachelor of Science in Mathematics-Economics. In August of 2013, he entered the Graduate School at the University of Texas at Austin.

Address: tecuth@gmail.com

This manuscript was typed by Thomas E. Cuthbertson.